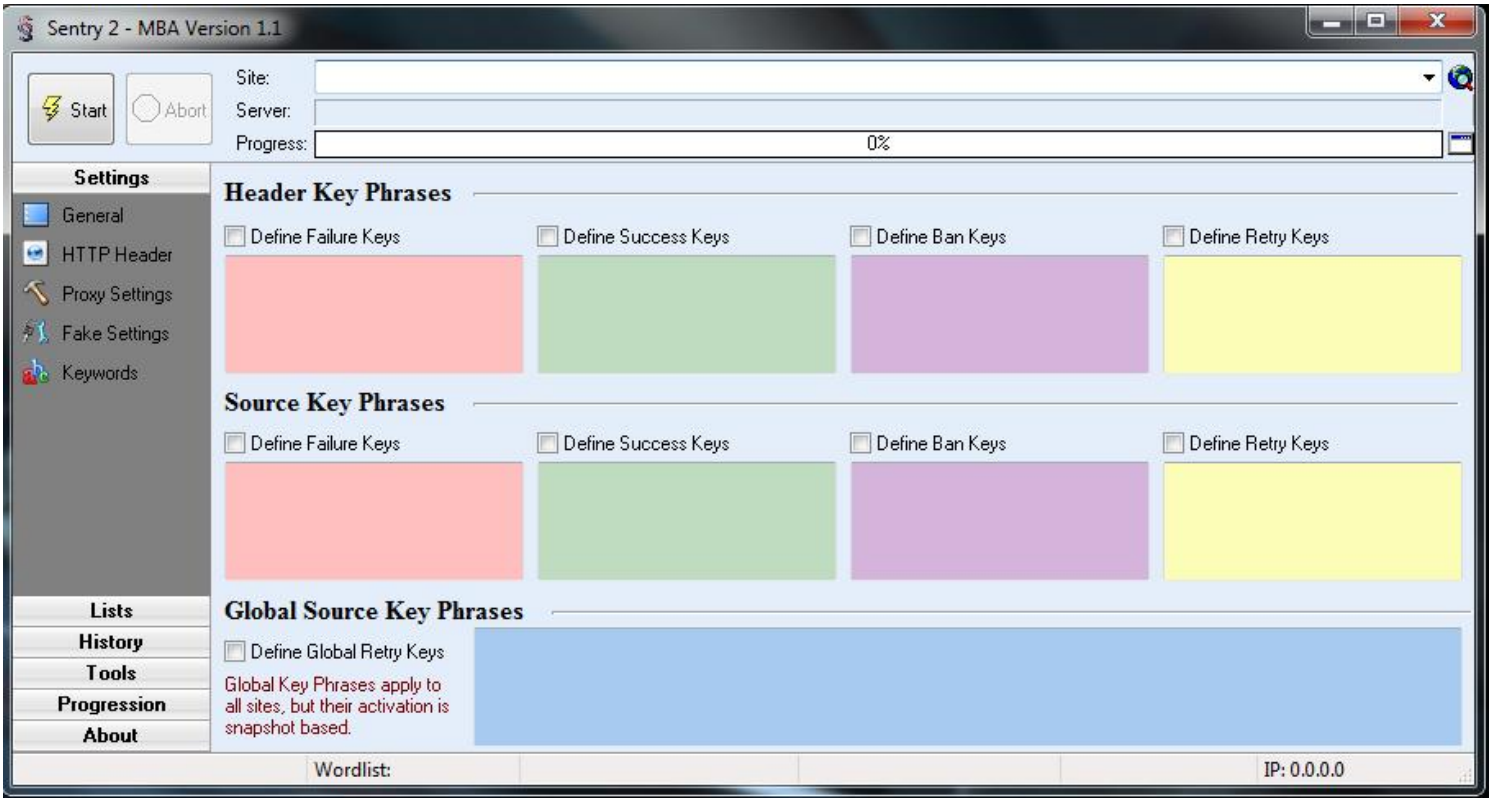
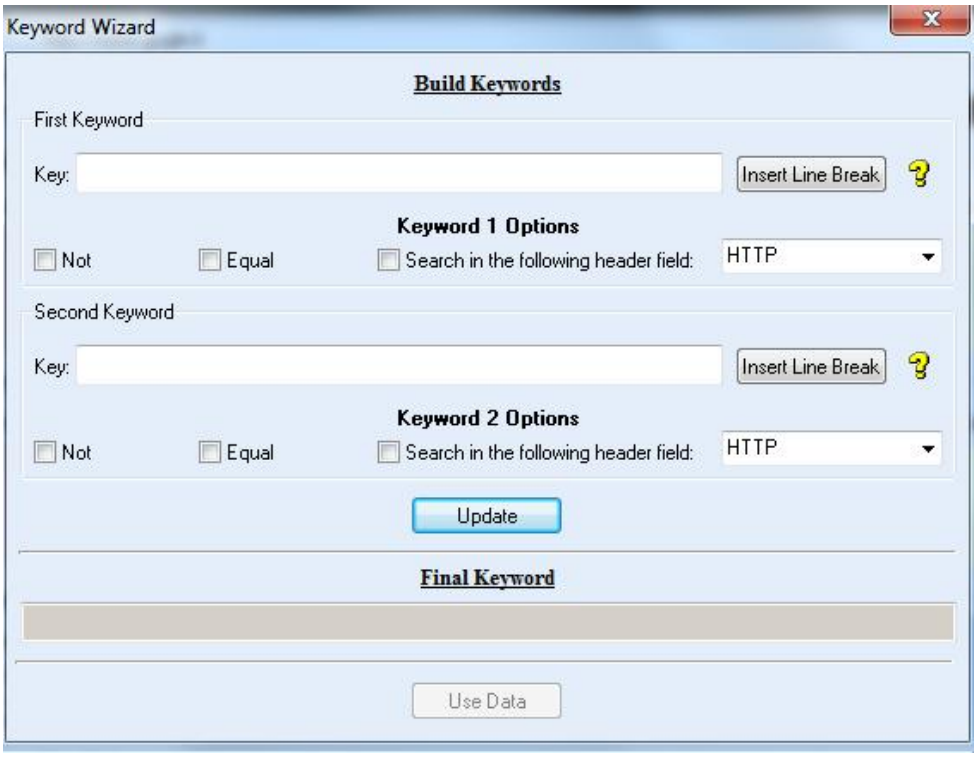
General Changes

Keyword Engine

The keywords phrases for Header and Source are now extended to four categories: Success Keys, Failure Keys, Ban Keys and Retry Keys. The keywords checker engine looks out first for header



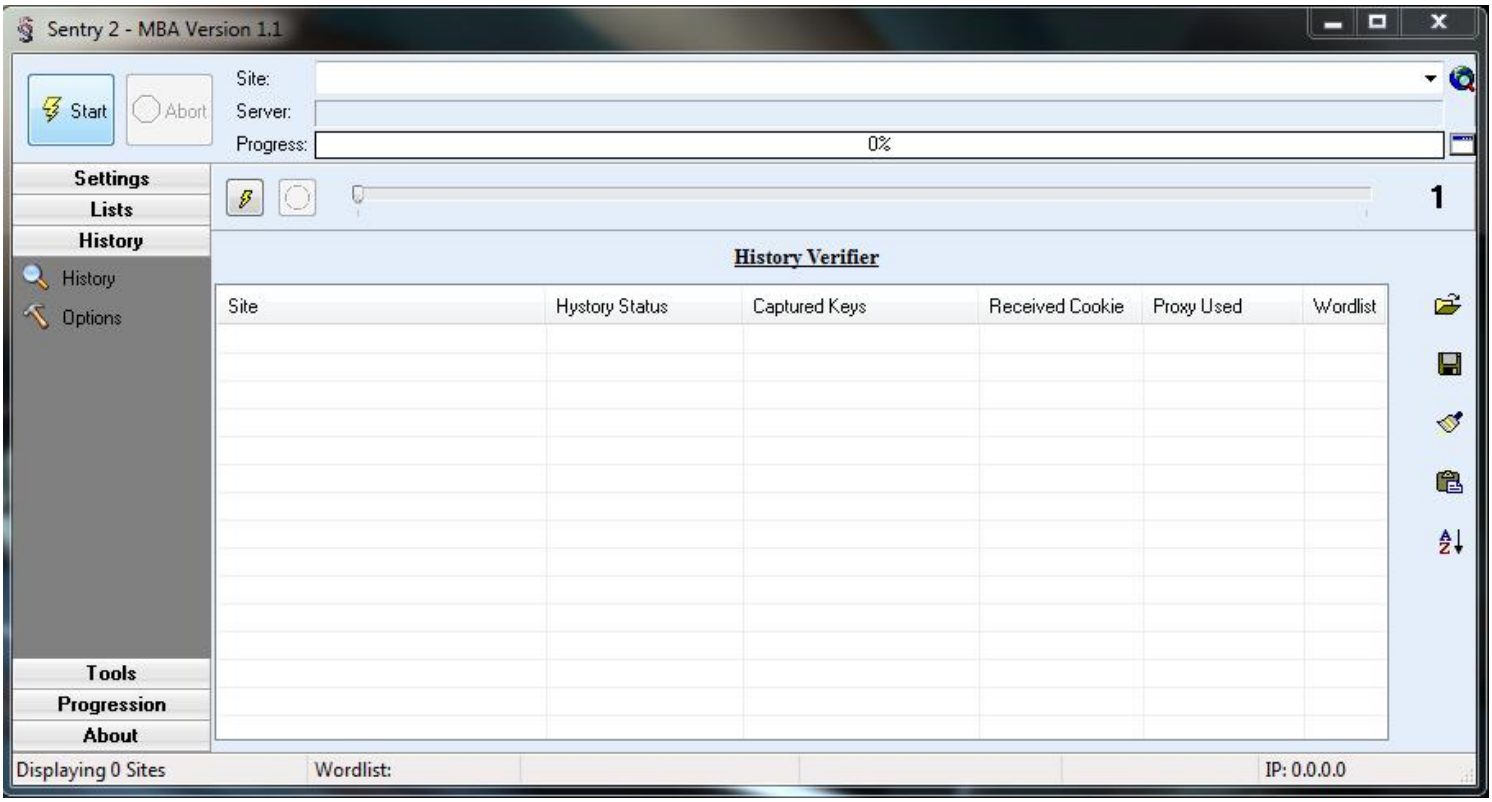


keys and then for source keys (the source keys are checked only if the request method is different from HEAD of course). It searches in order for Success, Ban, Failure and finally Retry keys. If a

success key is found, the combo is marked as hit. If a ban key is found, the current proxy is banned and the combo is tested with another proxy. If a failure key is found, the combo is marked as bad. If a retry key is found, the combo is retested with another proxy. If no keys are found, the AfterFingerPrinting Engine is activated. Moreover the global keys category has been changed to Retry Keys (a default list is included).
Finally you can access advanced key matcher features by right clicking on a key phrases box and selecting 'Add (Advanced)' from the popup menu. The Keyword Wizard will be launched: from here you can configure special matching functions.

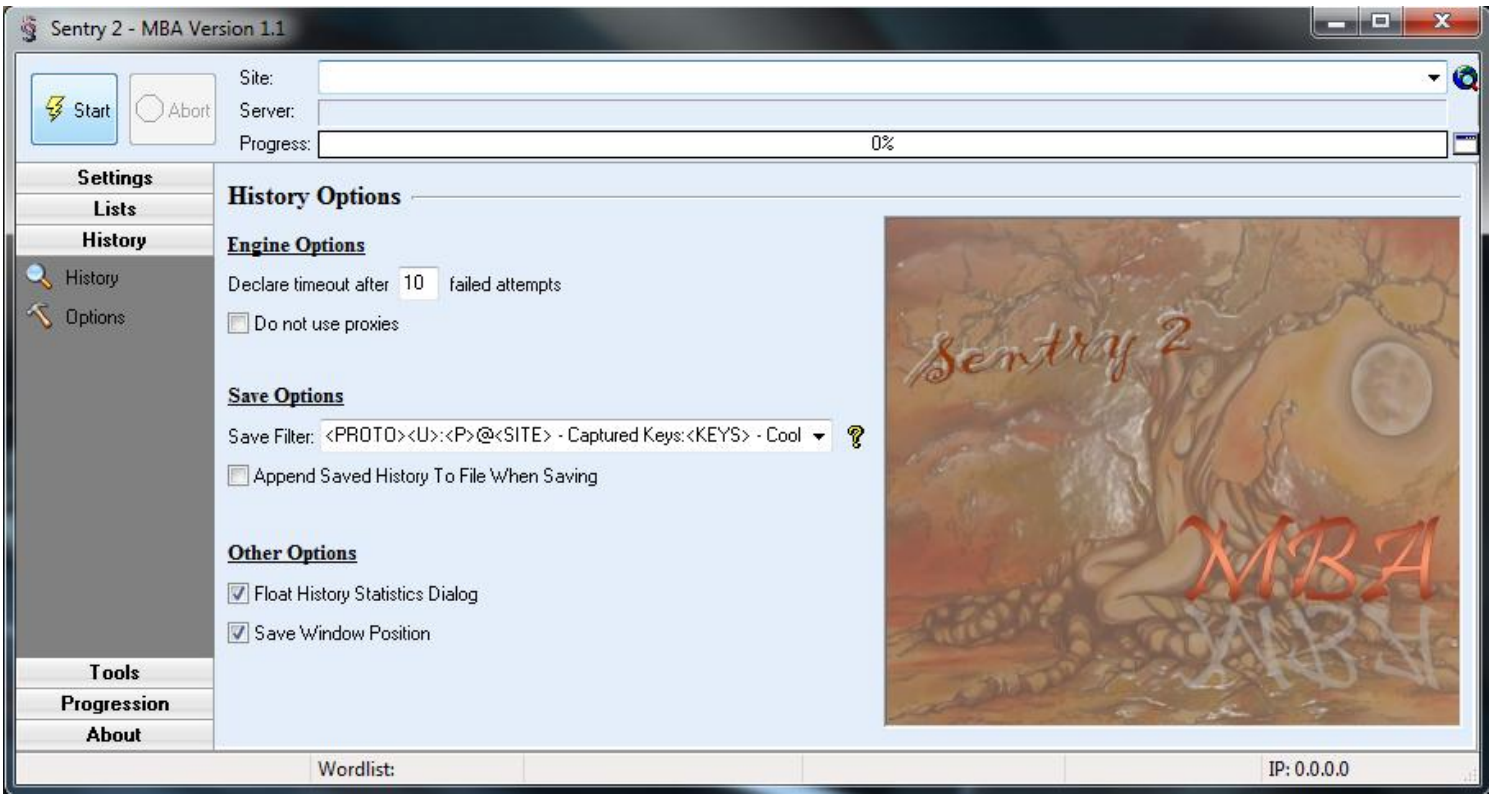
History Checker Engine

Now the History Checker engine uses the main bruteforcer engine to check history hits, i.e. for a given site in the history it uses the settings loaded from the site snapshot, so you don't have to configure anything in the history options, since all the checker settings will be acquired from a saved snapshot, unless you don't have a snapshot for the site...in this case you have to create one by starting a bruteforcer session with the right settings for the site you have to create a snapshot for.
The History list has 6 columns. The new columns are:



- Captured keys: this column shows the keywords captured on the answer page for each Hit. This column is refreshed each time you check a combo with the History checker.
- Received Cookie: this column shows the Cookie received by the target Site for each Hit. If the target Site is a hosting site and the combo is the login/password of a premium account, the received Cookie is a premium account. You can copy the Cookie by right clicking on the relative combo and selecting the new option "Copy selected received Cookies to the clipboard". This column is refreshed each time you check a combo with the History checker.
Moreover if you right click on a given site, you have a new option: "View Source Answer in Browser". By selecting this option, Sentry will load in your default browser the HTML source answer the selected site gave when the combo was marked as a hit by the bruteforcer engine or by the History Checker if the combo has been verified.
A combo in the history list will be first checked with the proxy that found the hit and, if the proxy fails, another proxy will be selected from your proxy list, until the engine terminates the test for the

given combo or the number of failed attempts becomes greater than the configurable value "Maximum Failed Attempts" you entered in the "History Option" frame (default value is 10).
In the "History Option" frame two new variables have been added to the Save Filter: <Keys> and <Cookie>. Upon saving to a file a Hit in the History list, <Keys> will be replaced with the Captured Keys string of the combo and <Cookie> with the received Cookie of the combo.
Finally you have the option to disable the use of proxies, i.e. a hit will be checked with direct



connection.

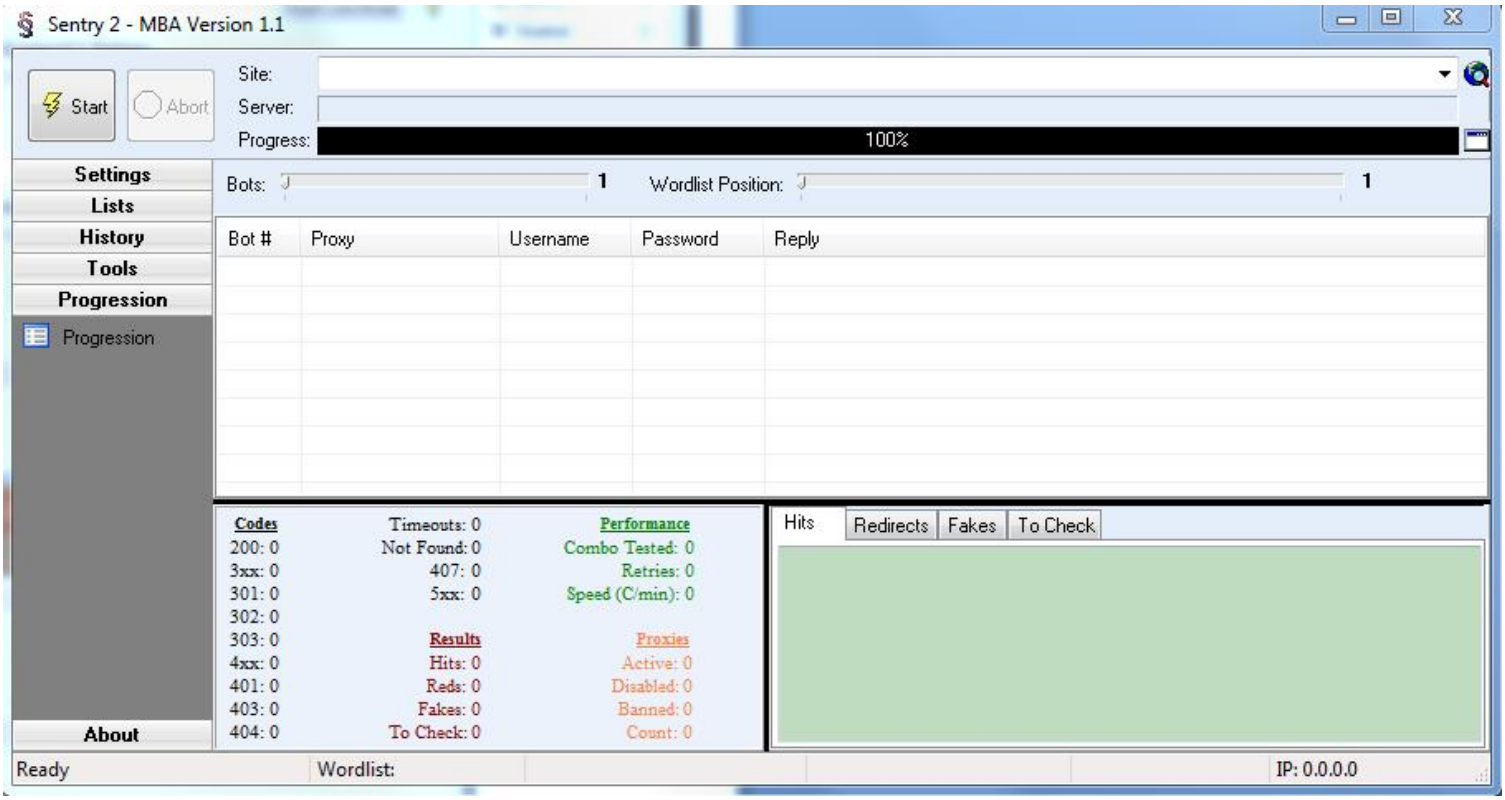
Proxies Analyzer Engine

Now the Proxies Analyzer Engine is a full 3 stages engine:
Check Proxy with ProxyJudge -> [Check HTTP] -> [Check HTTPS]
The stages between [] are optional.
A proxy is marked bad or timeout based on the result of the first stage. A bad proxy should be deleted from the list, since it gives bad codes (3xx, 401, 403, 407) or, with a 200 code, does not transmit the judge HTML source requested.
You can configure the max number of retries of a proxy before it is marked as timeout under the options form. The retries refer to the first stage attempts.
Finally new options have been added to the analyzer. The most important feature is the support of IpFilter.dat, a file that contains dangerous proxy ranges. An IpFilter.dat file is included with the MBA version. You can filter the proxy list by clicking on the 'Cleaning Options' icon and then selecting "Filter Proxies with IpFilter".

Progression Engine

Now the progression list in the progression frame gives more details on the status of the running bots, showing also the remaining time the bot has until timeout. Moreover the statistics panel too gives away more information. In particular now it shows the total combo tested form the start of the bruteforcer session and the speed, computed as number of combo tested in the last minute. Moreover a new tab has been added to the Progression Frame, labeled as "To Check". This tab show the detected "Forbidden Combos" and "Not Found Combos" and is used only for basic auth sites. The "Forbidden Combo" concept will be explained later, under the Fake Settings Frame of

this help . About the other type, a combo is marked as a "Not Found Combo" when the same combo gives " 404 - Not Found" with two different proxies. This can be useful when you enter incorrectly the path for members for an auth site: in this case you will get 401 with wrong combo and 404 - Not Found with correct combo. So if you don't see Hits, but have a lot of "Not Found" Combos, maybe this is the reason. Finally if you right click on a combo in the Hit Tab or in the Fake Tab, you have the new option "View Source Answer in Browser", that acts exactly as the one in the History Frame. It can be useful to see the answer of a fake, since, based on the answer, you can decide to add the proxy to the blacklist.

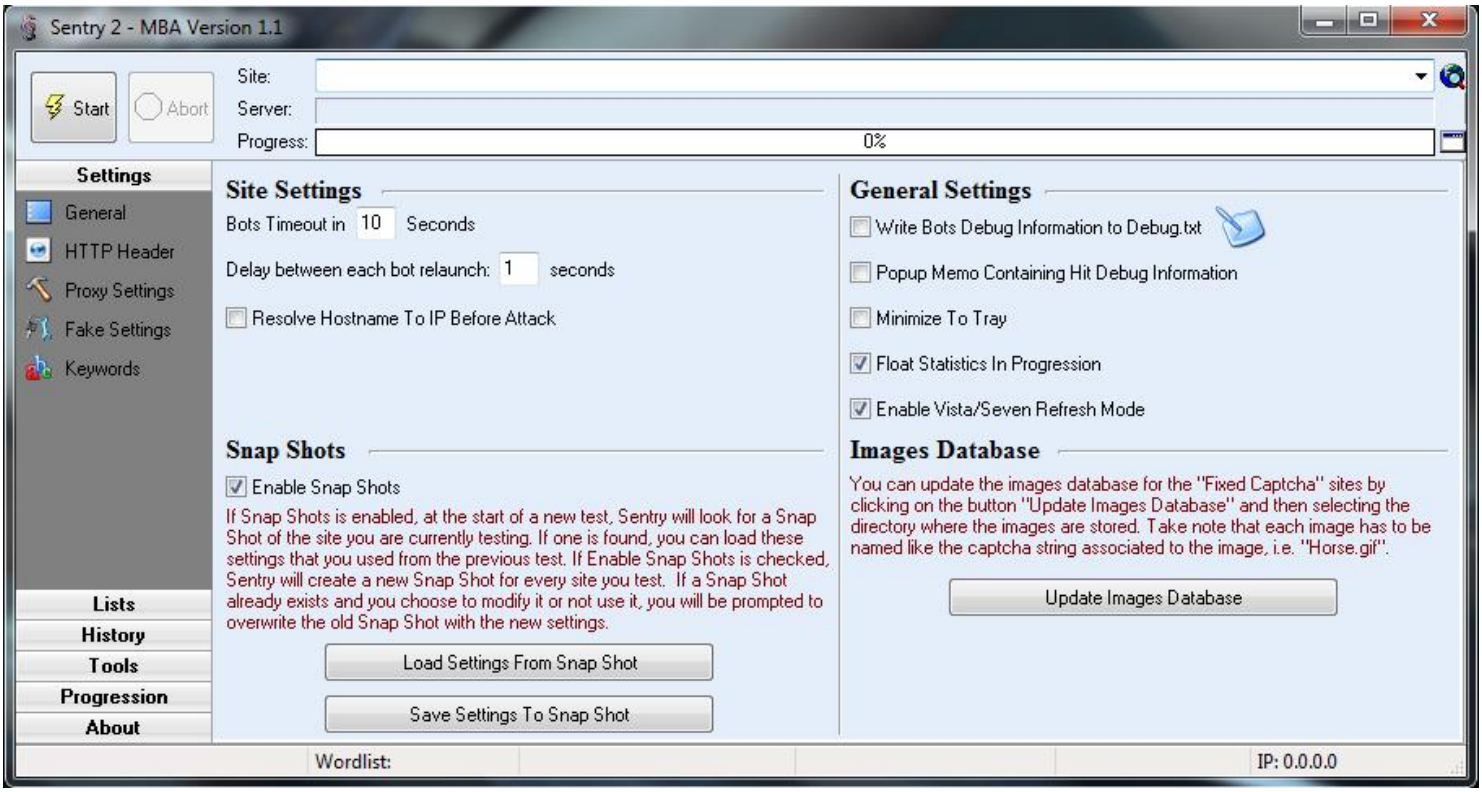


New Options

General Frame

- Delay between each bot relaunch

When a bot has terminated, the bruteforcer engine will wait X seconds before relaunching the bot . This can be useful for some sites that ban proxies based on a fixed number of failed attempts occurred in a fixed time interval.



- Enable Vista/Seven Refresh Mode

If you are on Windows Vista or Seven, just leave this option enabled.

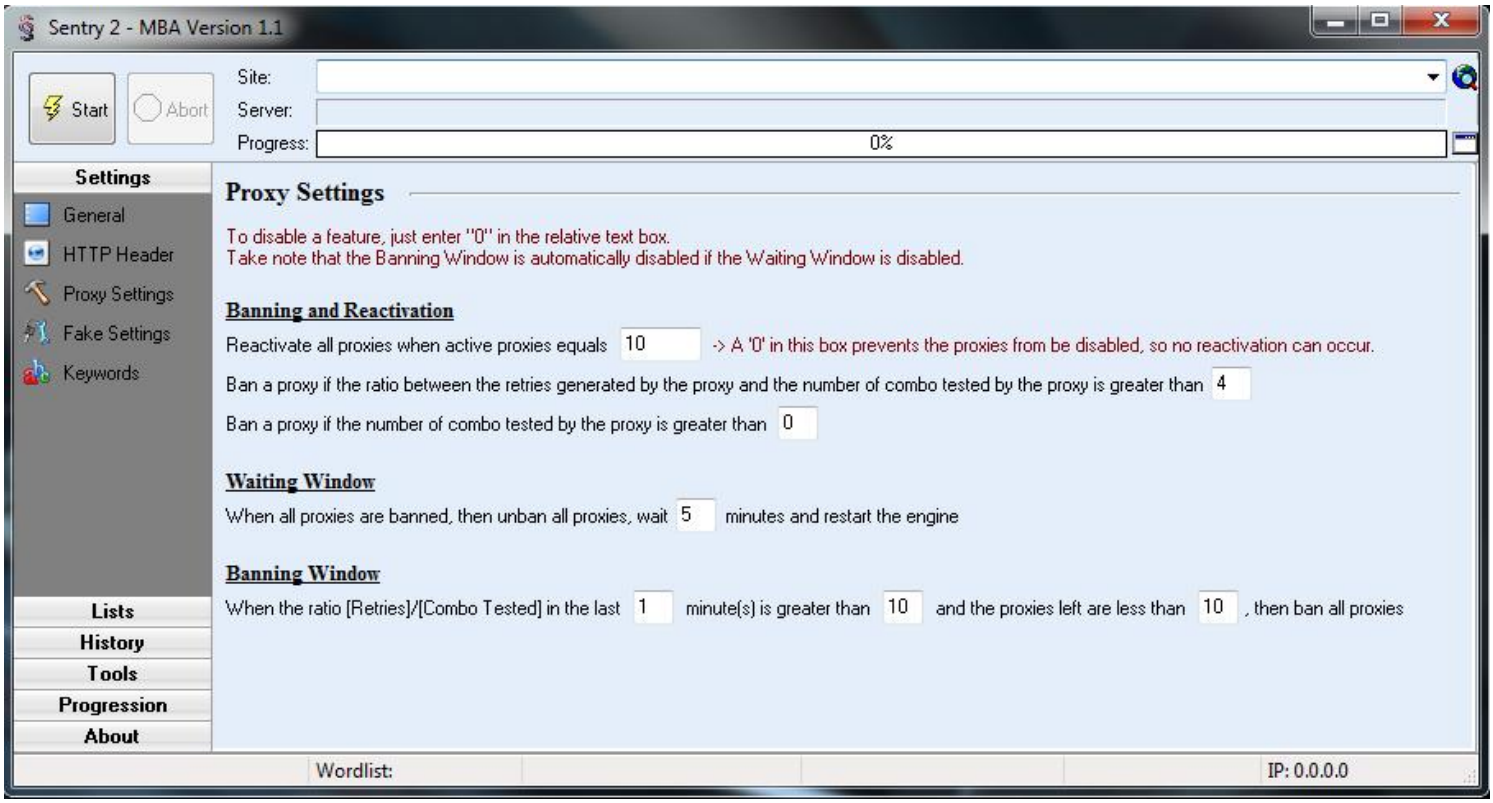
Proxies Settings Frame

Banning and Reactivation

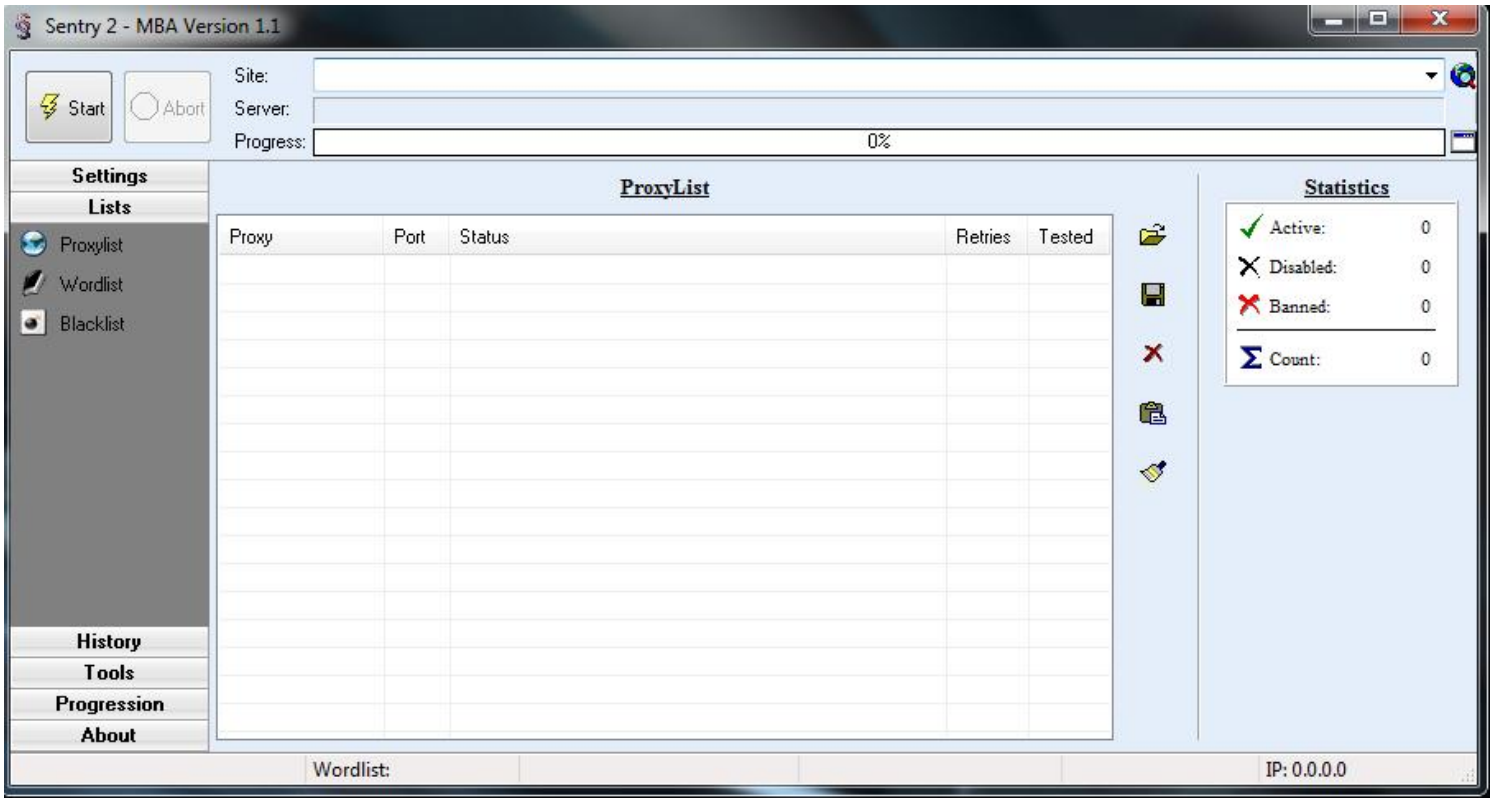
-The first option works like in the standard Sentry version, with the difference that if you enter "0", the proxies will not be disabled upon a proxy error. This can be useful for sites that ban proxies to avoid a quick banning of the fastest proxies. However this means that the slower or dead proxies are not disabled: this can lead to a slow cracking. To avoid this situation, a new option has been added.

-This option is labeled "Ban a proxy if the ratio between ...". Before explaining what this option does, I should add that the Proxylist now has two new columns: Retries and Tested. Retries are the number of Retries generated by the proxy, i.e. the proxy gave a bad HTTP code (5xx, 404 and so on) or a retry key match (retry key matches with OCR enabled are not computed, since they are usually linked to bad image recognition). Tested are the number of combo successfully marked by the proxy. These variables are used to ban a proxy that is marked as dead based on the ratio between Retries and Tested. The proxy is banned if this ratio becomes greater than the value specified in this new option. This feature can accelerate considerably the speed of the bruteforcing session when in the proxy list there are a lot of dead proxies. To disable this feature, just enter "0" in the text box.

-The next option is labeled "Ban a proxy if the number of combo tested by the proxy is greater than:". So what's the purpose of banning a proxy whose number of combo tested becomes greater than a specified value?



Well, the fact is that some sites upon banning a proxy continue to gives failures responses for both good combos and bad combos, i.e. with these sites it's not possible to recognize when a proxy is banned. But if we know the number of failed attempts after which the proxy is banned (it's easy if we know a valid combo), we can enter this number in this option. So what when all proxies are banned? Simple: the waiting window and the banning window will take care of the rest. This feature is disabled when "0" is set in the relative box (default).



Waiting Window

-When all proxies have been banned, the engine will stop and wait the amount of time specified before reactivating. For sites that ban proxies, by selecting an appropriate waiting time, most of the proxies will be unbanned upon reactivation. So the bruteforcing can goes on without user intervention. Enter "0" to disabled this feature.

Banning Window

-Let's suppose that most of your proxies have been banned and the ones left give only bad replies. Normally these proxies would been banned by the Proxy Engine based on the ratio between Retries and Tested, but it's not rare that a proxy is alive upon the start of the bruteforcing session and then starts to act like dead. In this case it would take forever for the Proxy Engine to ban such proxies. This feature solves such cases, as it forces the waiting window activation when the proxies left are less than a fixed number (user configurable) and the ratio between Retries and Combo Tested is greater than a fixed number (user configurable) in the last configured time interval. The banning window is disabled if you enter "0" in one of the three text boxes

Fake Settings Frame

-The AfterFingerPrinting Engine for basic auth sites now validates also 3xx codes (i.e. redirects) and 403 codes. The reason to validate 403 codes depends on the fact that some sites give 403 code for shared combos, so, before banning a proxy for a 403 code, the engine sends a wrong combo with the current proxy: if the answer is still 403, then the proxy is banned, otherwise the combo is marked as a forbidden combo and added to the new "To Check" tab in the progression frame. The AfterFingerPrinting engine can be also configured with the minimum number of success replies a bot must acquire for the current combo before terminate successfully the AfterFingerPrinting process, i.e. declare hit for a 200 code, redirect for a 3xx code and forbidden combo for a 403 code. A success reply is a 200, 3xx or 403 code with the current combo and a 401 reply with a wrong (randomly generated) combo. The number of success replies that trigger the success of the engine can be entered in the new option "Assume success when total good replies are" available in the "Fake Settings" Frame. Finally the AfterFingerPrinting engine is also enabled for form sites, although it cannot be disabled or configured for such sites.

-The option for the content length checker now applies for form sites too. Moreover the option

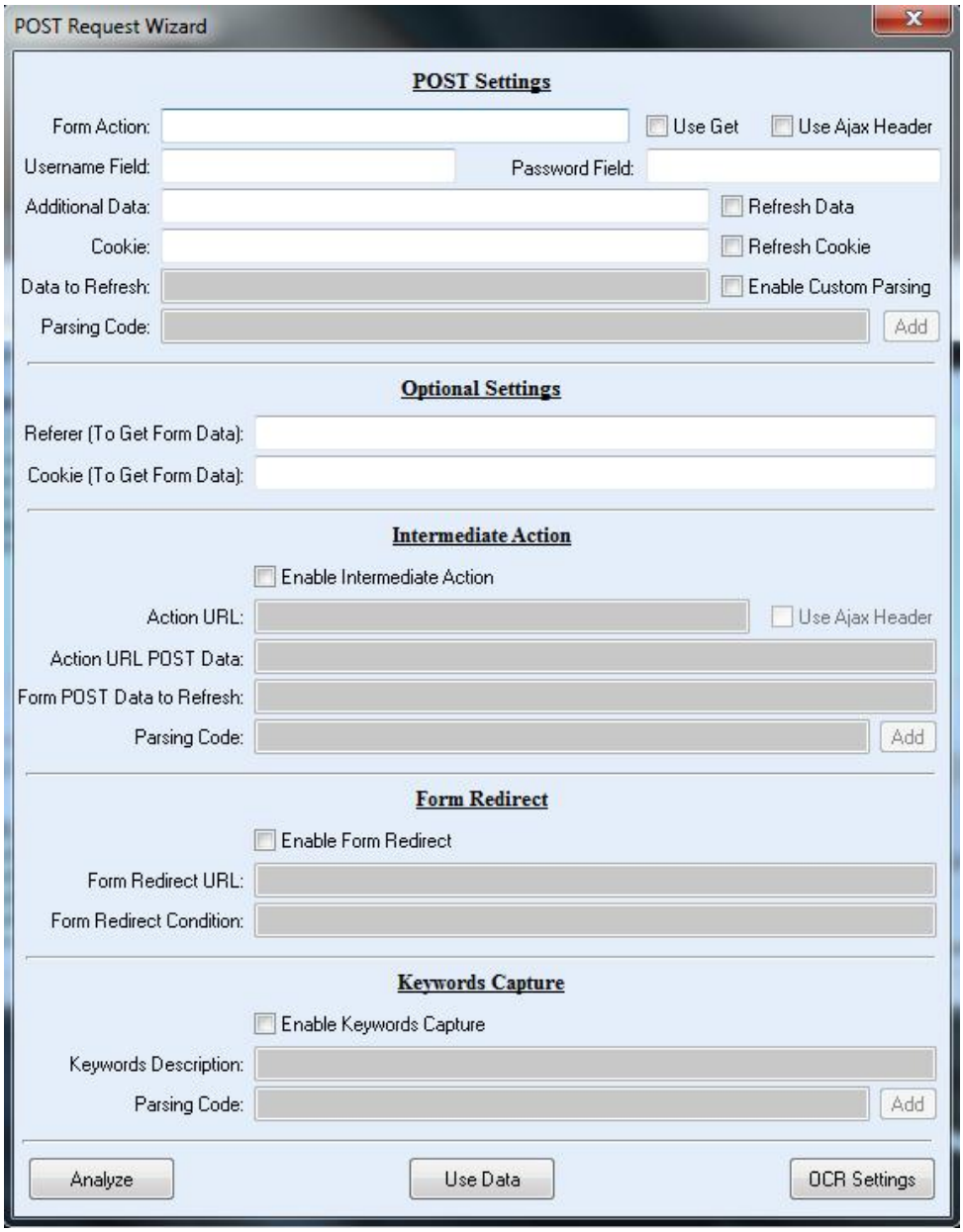
behavior is different. If this option is checked, when the bot engine receives an HTML source on which no keys are found and the length in bytes of the received source is less than the one specified, then a' 404 - Incomplete Source' error is issued and the combo is retested. This option is useful since most bad proxies give an incomplete source along with a 200 code: on such sources, even if the combo is bad, no keys would be found and the AfterFingerPrinting Engine would be activated. In this way we prevent a bad combo to be marked as hit.

Post Wizard Form

Before illustrating the new options, let's say that now the form engine is an up to 6 stages engine: [Get Login Page] -> [Get Captcha Image] -> [Call Intermediate Action] -> Post -> [Call Redirect URL] -> [Capture Keys]
The stages between [] are optional.
Get Login Page stage
In this stage, the form engine gets the login page to refresh cookie and or session data fields. It can also capture data fields with a custom parsing code
Get Captcha Image stage
In this stage, the form engine get the Captcha image that will be processed by the OCR engine to generate the Captcha Code.
Call Intermediate Action stage
In this stage, the form engine call an intermediate URL in order to gather additional post data fields with custom parsing code and/or to initialize correctly the post script
Post stage
In this stage the form engine sends the login information based on the actual combo
Call Redirect URL stage
This stage is triggered when the post answer contains an user defined key. If such key is found, then the redirect URL (also user defined) is called and the source keys are tested against the response
Capture Keys stage
If the answer from the Post Stage or from the Call Redirect URL stage contains a success key, the HTML source of the answer can be parsed for capturing selected strings with a user defined parsing

Post Settings

-Use Get
If this box is checked, the Form Action URL will be called with GET instead of POST. Useful for some Ajax scripts that require this method.
-Use Ajax Header
If this Box is Checked, the form engine will add " X-Requested-With: XMLHttpRequest" header to the headers sent to the form action. This is required by some Ajax scripts.
-Refresh Data
If this Box is Checked, the form engine will get the login page to refresh ONLY the additional data fields.
-Refresh Cookie
If this Box is checked, the form engine will get the login page to refresh the cookie. If you enter a cookie with this box selected, the cookie will be sent together with the cookie received from the login page. If the name of the cookie you enter is the same as the one received from the login page, your cookie will be overwritten with the refreshed one.
-Enable Parsing Code
If this box is checked, the form engine will gather additional fields from the login page by parsing the HTML source received with your parsing code. The fields extracted will be added to the ones entered in the additional data text box, that will act as fixed fields. This option is useful when the default parsing engine is not able to refresh the additional fields, i.e. fields are generated by JavaScript. Let's explain how this work.
The syntax is the following:



Data to Refresh: <Field_Name>
Parsing Code: <Left_String>|<Right_String>
where <Field_Name> is the name of the field you want to capture, <Left_String> is an unique string in the HTML source after which the value string you want to capture starts and <Right_String> is a string (usually a character is enough) that comes after the last character of the value string.
You can match line breaks and tabs with **\n** and **\t** respectively.
Now what if you want capture more than one field?
No problem. The syntax is the following for two fields:
Data to Refresh: <Field_Name1>&<Field_Name2>
Parsing Code: <Left_String1>|<Right_String1>&<Left_String2>|<Right_String2>
Anyway you can use the Parsing Code Wizard to generate automatically the parsing code. To launch the Wizard, just click on the "Add" button. An example will be given in the "Keywords Capture Settings" section.

Intermediate Action Settings

-Use Ajax Header

If this Box is Checked, the form engine will add " X-Requested-With: XMLHttpRequest" header to the headers sent to the intermediate URL. This is required by some Ajax scripts.

If "Enable Intermediate Action" box is checked, then the Form Engine will call the URL specified in the "Intermediate Action URL" text box before posting. If the box "Intermediate Action POST Data" is leaved empty, then the calling method will be GET, otherwise POST. This feature is important for Ajax sites that require the Ajax script to be initialized before posting. Moreover you can grab additional data fields to be added to the ones specified in the POST Settings by configuring the text boxes "Form Post Data to Refresh" and "Parsing Code". The syntax for these fields is the same described for "Data to Refresh" and "Parsing Code" text fields in POST settings. Of course you can use the Parsing Code Wizard in this case too.

Form Redirect Settings

If "Enable Form Redirect" box is checked, then the Form Engine will call the URL specified in the "Form Redirect URL" text box after posting if the string specified in the "Form Redirect Condition" text box is found on the HTML source answer. Then the answer from the Form Redirect URL will be checked against success, failure, ban and retry keys.

This can be useful for certain hosting sites that redirect by JavaScript to the real account page when a successful combo is posted. In this case the Form Redirect URL would be the URL of the account page and the Form Redirect Condition would be for example the JavaScript code that redirects to the real account page. By enabling this option, upon getting the real account page, you can for example differentiate between free and premium accounts with the correct success keys.

Keywords Capture Settings

If "Enable Keywords Capture" is checked, then the form engine will try to capture strings on an HTML source where a success key is found. The" Keywords Description" text box contain your personal description to the captured strings, i.e. Premium Until, Traffic left and so on. The "Parsing Code" text box will define how the capture engine must capture the strings. The syntax for this two boxes is the same one described for "Data to Refresh" and "Parsing Code" text boxes in POST settings. Let's make an example by using the Parsing Code Wizard.

Let's suppose you have this string in the HTML Source answer:

Premium (311 days remaining - extend)

This comes from a well known hosting site. Follow these steps:

1) Check the box Enable Keywords Capture

2) Launch the Parsing Code Wizard by clicking on the Add button

3) Paste the HTML Source above in the big box under the Find Button

4) Now select 311 and release the mouse left button while holding shift: the wizard will generate automatically the left and right strings

5) In the field name text box enter your own description for the captured strings, i.e. "Premium Until"

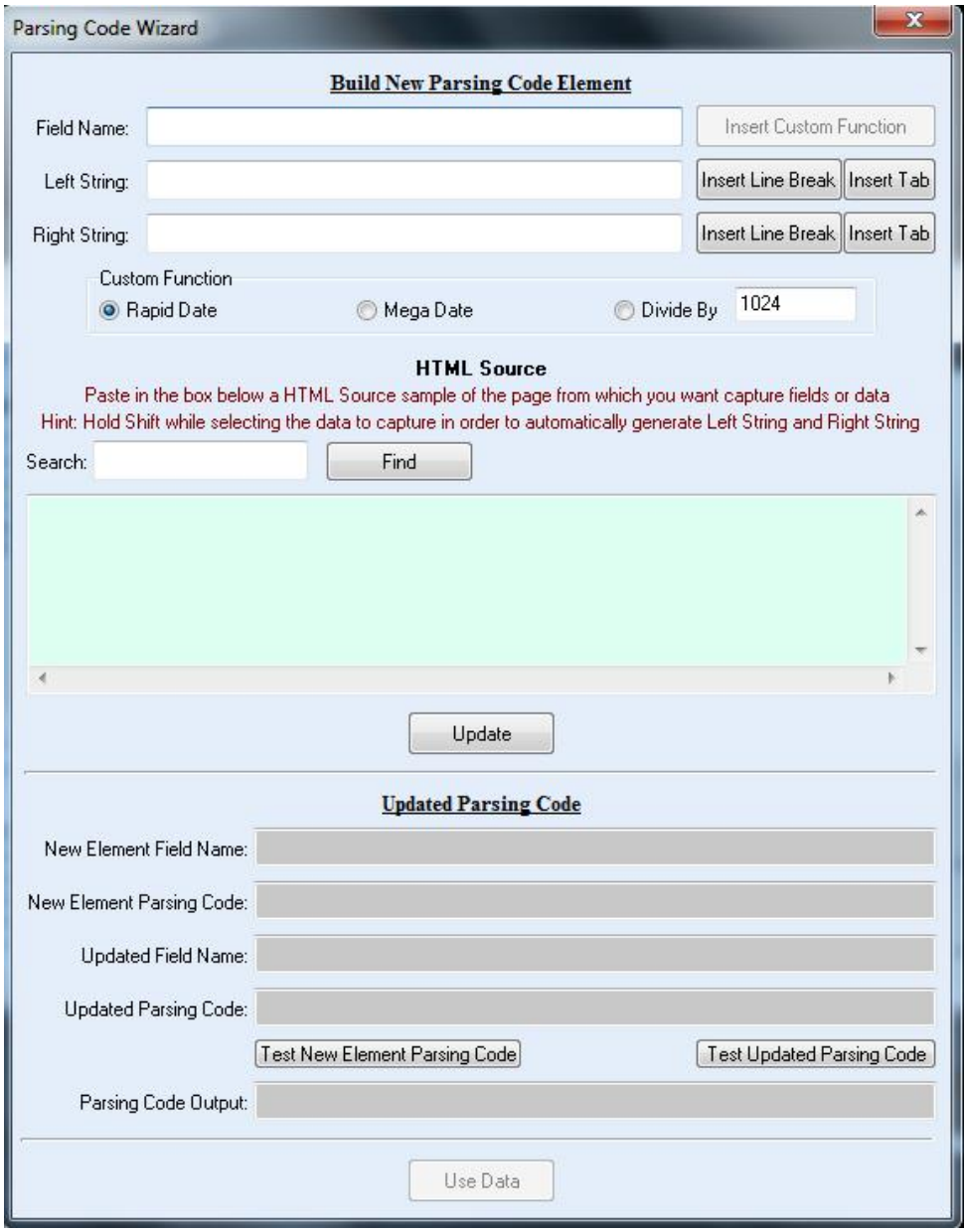
6) You can process the string captured with the following Parsing Functions:

- Mega Date: Process a relative date given in day remaining until a membership expires and returns the date in absolute time. This is useful for Megaupload and other sites.
- Rapid Date: Process a date given in Unix format and returns an absolute date. This is useful for Rapidshare and other sites.
- Divide By: It divides the string captured by X and returns the result. This is useful for the traffic left on premium accounts when we want the traffic in GB, but the site gives the traffic in B or MB.

In this case let's select Mega Date.

7) Click on Update and then on "Test new element parsing code".

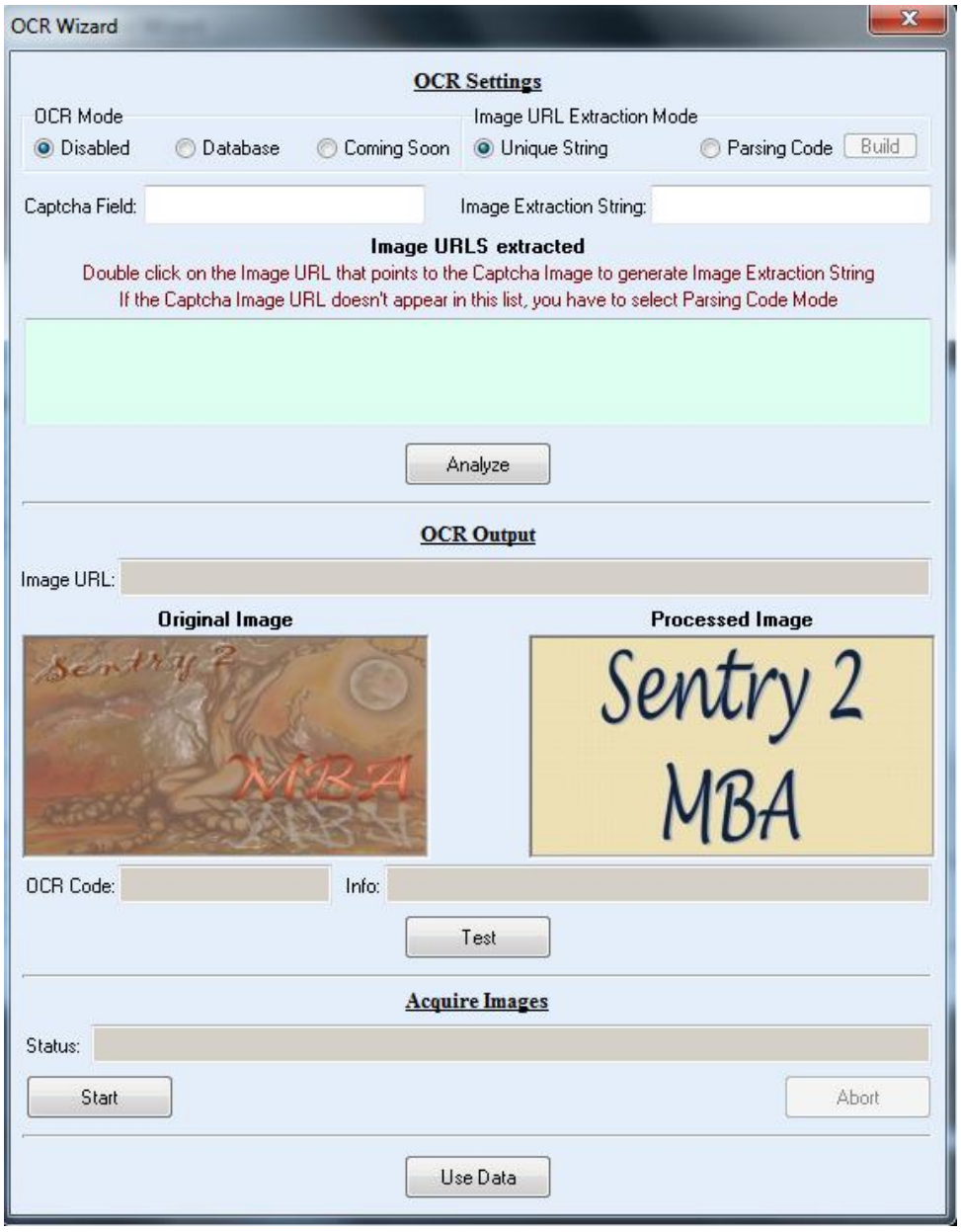
8) When you are happy with the settings click Use Data.



9) If you want to capture another string just click on Add. You can capture how many strings as you want.

OCR Wizard Form

From this window, you can configure all OCR related settings.
First, you have to enable the OCR engine if the site requires OCR processing. ATM you can select only database mode, i.e. the image recognition will be based upon the database file image.dat, included with the MBA version. You can enable this operation mode for sites that use fixed captcha image, i.e. not real-time generated. Some sites are included in the image.dat file, but you can add other sites by using correctly two features of the MBA version. See next paragraph for details. Once you have enabled the OCR engine, press the button 'Analyze'. If you have set up correctly the login page URL, the extracted image URLs box will be populated with all image URLs found on the login page. From this list you should recognize the URL that points to the captcha image: double click on it and you are set. If the captcha image URL is not present in the list, then the desired URL is likely JavaScript generated. In order to extract the image URL, select 'Parsing Code Mode' from the URL extraction options, then click on 'Build': the parsing code wizard will appear. In the HTML source box, find the captcha image URL, select it while holding shift, then click on the 'Update' button and finally on the 'Use Data' button.



Now it's time to test the OCR settings: click on the 'Test' button: the captcha image should appear in the 'Original Image' box, if not, then some settings are wrong. If the correct image appears, but it is not recognized, then the image is not present in the database. If the image is correctly recognized, then click on the 'Use Data' button.

How to add images to the database

First, we have to make sure the captcha system for which we want to add images to database use fixed images. So, let's configure correctly the OCR settings as described previously. Now in the OCR Wizard let's click on the button 'Start' under 'Acquire Images'. Take a look at the Status box: if the number of duplicate images increases as the process goes on, then the site uses fixed captcha images. In this case wait until the number of downloaded images doesn't increase anymore: at this point you should have acquired all the captcha imagse and can abort the acquiring process. Go in the directory where the images have been saved (Images in Sentry root) and rename each image downloaded with the associated captcha code.

Finally open the general setting frame in Sentry, press the button "Update Images Database" and select the directory where you put the saved images (make sure that only the renamed images are in this directory) and click 'Ok'. At the end of the process, Sentry will tell you how many images have been added to the database.